

LISTing Newsletter

Newsletter of the Long Island
Sinclair/Timex Users Group

December 1993 Issue
NEXT MEETING DEC 12, 1993



TABLE OF CONTENTS ***** ** *****

PG 3, QL CORNER

PG 4, TASWORD PLUS

PG 7, CUSTOMIZING YOUR WORD
PROCESSOR

PG 10, SPACE RESCUE



Listing Policy

Annual Dues \$16.00

One "sample" copy sent upon receipt of Business size SASS.
Copies provided on EXCHANGE BASIS with other bona fide user
groups. LISTing is published monthly except July and August by
LIST (Long Island Sinclair Timex) Group, a not for profit user group.

We are always looking for articles, programs, reviews etc. to keep our
members informed and entertained. You maintain full credit and copyright.

Portions if this publication may be reproduced, without written consent.
Please give credit to LIST when reprinting articles.

LIST disclaims any responsibility for any damage you may do to your
computer as a result of reading any articles in LISTing.

 LIST OFFICERS

 PRES. HARVEY RAIT
 U.P. BOB GILDER
 TRES. ROBERT MALLOY
 COR. SEC. JOHN PAZMINO
 EDITOR. FRED STERN
 LIBR. TOM SKAPINSKI



PLEASE SEND INQUIRIES TO:
 LIST
 MR. HARVEY RAIT
 5 PERI LANE
 VALLEY STREAM, N.Y. 11581

PLEASE SEND SUBMISSIONS TO:
 LISTING
 MR. FREDERIC STERN
 P.O. BOX 264
 HOLBROOK, N.Y. 11741

COMING EVENTS:

 DES. 12, 1993 LIST MEETING.

 SPECIAL NOTICE

THE NEXT MEETING WILL BE HELD AT
 THE ICE CREAM DISPENSARY
 (HARVEYS STORE)
 334 DOGWOOD AVENUE
 FRANKLIN SQUARE, N.Y.
 TEL: 516-486-1090

DIRECTIONS: SOUTHERN STATE PKWY.
 TO EXIT 17 NORTH (HEMPSTEAD AVE.)
 GO TO FIRST TRAFFIC LIGHT,
 LEFT TURN ON TO CORNWALL,
 NEXT TRAFFIC LIGHT, BEAR RIGHT
 ON TO DOGWOOD AVENUE. GO 1 MILE
 TO THE ICE CREAM DISPENSARY, IN
 A SMALL SHOPPING CENTER ON THE
 LEFT SIDE OF THE ROAD.

MEETING MINUTES

 REPORTED BY: FRED STERN
 NOV. 14, 1993

 HARVEY CALLED THE MEETING TO
 ORDER AT 2:45PM.

WE RECEIVED AN ENQUIRY ABOUT
 LIST, AND A MEMBERSHIP RENEWAL
 IN THE CORRESPONDENCES.

BOB GILDER HEARD FROM JOE
 LA PUNZINA, WHO HEARD FROM BOB
 DYLE THAT I.O.L.R. WILL BE
 SPONSORING ANOTHER QL SHOW IN
 NEWPORT R.I. NEXT MAY. DETAILS
 WILL BE PUBLISHED AS WE RECEIVE
 THEM.

LATE BULLETIN - WE HAVE
 RECEIVED WORD THAT JOHN PAZMINOS
 MOTHER HAS PASSED AWAY. WE AT LIST
 EXTEND OUR SINCERE CONDOLENCES
 TO JOHN AND HIS FAMILY.

 * LIST * LIST * LIST * LIST *

 * LIST * LIST * LIST * LIST *

 * LIST * LIST * LIST * LIST *

TECHNICAL ROUNDTABLE *****

A DISCUSSION WAS HELD ON NOTCH
 PUNCHING 720K DISKS TO FORMAT
 THEM AS 1.4MEG DISKS.

THE CONSENSUS BY MEMBERS WHO
 HAVE DONE THE PROCEDURE WAS
 FAVORABLE. THE 720K DISKS MOD-
 IFIED FOR 1.4MEG USE HAVE BEEN
 USED SUCCESSFULLY ON QLS WITH
 THE COMPATABLE 1.4MEG DISK DRIVE
 WITH NO PROBLEMS.

SWAP MEET IS COMING.
 **** **

LIST WILL BE HOLDING A JANUARY
 SWAP MEET WITH A TWIST.
 IN ORDER FOR OUT-OF-TOWN MEMBERS
 TO BE ABLE TO PARTICIPATE, THIS
 YEAR WE WILL DO SOMETHING NEW.
 SEND A LIST OF EQUIPMENT THAT
 YOU WANT TO SELL OR BUY OR SWAP.
 AT THE JANUARY LIST MEETING ALL
 THE LISTS WILL BE COLLECTED AND
 PUBLISHED IN THE FEBRUARY ISSUE
 OF LISTING. ALL MEMBERS WILL
 THEN HAVE THE OPPORTUNITY TO BUY
 OR SELL THE EQUIPMENT THEY
 DESIRE.
 ON THE EQUIPMNT LIST YOU SEND
 PLEASE INCLUDE;
 YOUR NAME
 YOUR ADDRESS
 QUANTITY AND TYPE OF ITEM
 PRICE FOR ITEM YOU ARE SELLING
 SELLING OR SWAPING
 WHAT YOU WILL SWAP FOR
 WHAT YOU WANT TO BUY

CLASSIFIEDS

 THIS CLASSIFIED SECTION IS
 AVAILABLE TO ALL LIST MEMBERS
 FREE OF CHARGE.
 THE ONLY RESTRICTION IS THAT
 IT IS TO BE USED ONLY FOR THE
 SEEKING, SELLING OR SWAPPING
 OF SINCLAIR, TIMEX OR MICROACE
 COMPUTER EQUIPMENT, PERIPHERALS
 AND SOFTWARE.
 LISTING, LIST, AND ITS OFFICERS
 DO NOT ENDORSE, WARRANTY, OR
 GUARANTEE ANY OF THE ITEMS
 LISTED IN THIS CLASSIFIED
 SECTION

 THE FOLLOWING PUBLICATIONS ARE
 AVAILABLE ONLY THROUGH LIST:

ZX-81/TS1000 TECHNICAL TIDBITS
 TECHNICAL TIDBITS PART II
 SAVINGS AND LOAD OF THE TIMEX
 COMPUTER
 \$4.00 EACH.

I NEED THE DOCUMENTATION FOR THE
 FOLLOWING PROGRAMS USED ON THE
 TS1000:
 MINI-MOD 1.5
 Z-COMM
 FREDERIC STERN, LISTING EDITOR.

I AM LOOKING FOR DIRECTIONS OR
 DOCUMENTATION FOR THE ZX ASSEM-
 BLER AND DISASSEMBLER BY BOB
 BERCH.
 MR. WAYNE KNAUST
 2 PEAR TREE COURT
 ST. PETERS, MO. 63376

QL CORNER

This month's article has been inspired by Harvey Rait, President of the LIST Group. Harvey's QL power supply failed and he asked me if I could repair the power supply for him. The only problem that I had was opening up the supply case.

I was able to remove two out of the four, cross-slotted screws, destroying two screw drivers during that process. Fred Stern took the power supply home; drilled out the remaining screws and delivered the supply to me via Bob Malloy. This was a real team effort and this is just what LIST is all about!

The Power Supply consists of a small printed circuit board attached to eight transformer windings. First, I checked the output voltage from the line delivering 9 - 12 VDC to the internal QL regulator (the RED lead). The voltage reading was approximately +5 Volts DC, much too low! The first check was to determine if the four rectifier diodes were OK. Using an OHM meter I checked each of the four 1N5400 diodes, back to back. One of the diodes appeared to be shorted, so I removed all four of them and then checked them in both directions with an OHM meter. One diode was shorted and another diode displayed some leakage.

I replaced the two original good diodes and installed two new diodes (1N5401, 100 Volt rating) into the printed circuit board and then tested the output for proper voltage. The supply output indicated +11.5 VDC unloaded (with out the cable attached to the QL). The Power cable was then connected to a QL and when powered up the QL responded as it should.

Some words of caution about repairing the QL power supply: The printed circuit board is made from phenolic resin PC material with a thin copper overlay. This type of PC material is prone to warps and cracking. If you have to remove any of the SIX diodes on the board, use a hot soldering iron and some sort of desoldering tool to pry the diodes out. The diode leads are approximately 1/16" in diameter, heavier than most components you will encounter. Before you reinsert the original diodes into the PC board, carefully file off any burrs and excess solder coating on the diode leads so that they fit into the PC board holes easily - otherwise if you apply too much pressure on the diodes during insertion and the leads aren't smooth, the chance of lifting up the foil trace on the PC board is great, making not only the repair difficult but having to repair the broken trace by adding a short length of tinned wire across the fractured trace.

I tested Harvey's Power Supply unloaded, in other words, it was not connected to the QL before I was sure that the voltage for the QL internal voltage regulator fell between +11 - +12VDC. Once the proper unloaded voltage fell within the above stated range; then I disconnected the power cable from the AC source; connected the cable to the QL and then powered up the supply for testing.

I made an attempt to adjust the 2K2 (2.2K) variable control for a slightly higher voltage output. To my surprise, the control was sealed with a solvent preventing any adjustment of the control. The power supply was then reassembled into the case and I used 4-40x1/2" machine screws as a replacement for the original screws.

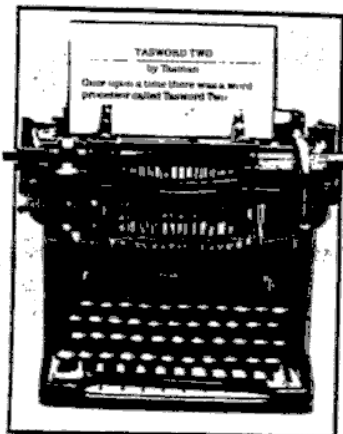
Next month I will supply the printed circuit board layout, a component placement layout, a list of components indicating both US and European equivalents and a schematic for the power supply. I will also list voltage measurement points at all component junctions on the PC component layout and if I have the room, I will attempt to provide a simplified theory for the QL power supply.

See you next month.... Bon Gilder

Tasword plus.

John Wall shows how to add Wordcount, Paragraph-count and Header facilities to Tasword II.

T52055 T52055



Word Count

Tasword Two has rapidly become the standard Word Processor for the Spectrum and has most of the features that purpose built WP's have, two features that are missing however are a current word count and automatic header. This is a machine code routine

that gives, in a fraction of a second, the number of words typed into the file up to present. Controlled from BASIC, it also gives a paragraph count.

The main problem is where to put the code. In Tasword Two the text file is held between 32000 and 52480 with an extra 128 bytes after that obviously used for overflow routines. The machine code section is held from 54780 to 65535. However, the machine code also uses bytes lower than 54780 for data storage. I chose 52610 as an address as far as possible from the data section of the machine code routines and no problems have arisen. The main disadvantage is the extra time the program takes to SAVE and LOAD, perhaps another 10 seconds each for the basic and the machine code.

The machine code routine involves only relative jumps. You might be puzzled by the 256 that is added to the DE register initially. This is so that the test for the end of the text file is

simplified. Just before the end is reached DE will hold 00 01 (hex — least significant byte first) and the next decrease of DE will leave FF 00 (255 decimal) and the D register will hold zero and the routine will return to basic. The word count is held in the BC register so that the command PRINT USR 52610 will return the actual count (See lines 60 and 9360). The code could be made shorter by omitting the CORRECT FOR END OF LINE ERROR routine. However this would give a false count because the routine would not separate two words one of which ends at column 64 and one which begins at column 1 of the next line. There is a brief explanation of the code in figure 1.

Header Routine

This enables the recall of a pre-entered address heading of up to seven lines. It also pushed down the entered text so that the

heading does not overwrite it. There is a facility from BASIC to change the heading at any time. The header information is stored between 52660 and 53107 — 448 bytes or seven lines of text — and the code to manipulate it is from 53110 to 53145. The code is three, almost identical, block transfer routines of 12 bytes each. The first is described in figure 2.

The second routine has the values in HL and DE interchanged. The third routine moves the already entered text seven lines down and uses the LDDR instead of LDIR.

The Basic at lines 9000 — simply calls the three routines in the right order. Routine three moves the text, then routine one prints the header. Routine two is used when a new or edited header is needed. See lines 9010 and 9040.

Figure 1. the machine code routine.

INITIAL CONDITIONS

Word count set to zero
File length + 256 20,736
File start address 31,999

LD BC, 00 00
LD DE, 51 00
LD HL, 7C FF

CHECK IF NEXT BYTE IS NOT A SPACE

Select next byte
How many bytes to go?

a INCL HL
DEC DE

CHECK IF FINISHED

Have we reached the end of the file? Is the most significant byte of DE zero?
If so return to basic.

LD A, 00
CP D
RET Z

Is this byte a space? If so try next byte.

LD A, (HL)
CP 20 32 dec = space
JR, Z next byte (a)

INCREASE WORD COUNT

If not a space then must be start of new word. Increase word count by One.

b INC BC

LOOK FOR END OF WORD

Select next byte of word.
How many bytes to go?

c INC HL
DEC DE

CHECK IF FINISHED

Have we reached the end of the file. Is the most significant byte of DE zero?
If so return to basic.

LD A, 00
CP D
RET Z

CORRECT FOR END OF LINE ERROR

Is this the end of a line?

LD A, 3F (63)
AND L
CP L

If not continue with next byte of word.

If last byte of line then check next byte — first of next line.

JR NZ, d
INC HL
LD A, (HL)
CP 20
DEC HL

If next byte is a letter then go to word increase.

JR, NZ, b

If this byte is not a space then try next byte of word.

d LD A, (HL)
CP 20 32 dec = space
JR NZ, c

IF END OF WORD GO TO START
If this byte is a space look for next word.

JR, a

Figure 2. Block transfer routine.

Load HL with address of first
byte to be moved:
Load DE with address of
destination:
Load BC with length of block:
Use LDIR
Return to BASIC:

```
LD HL, CD B4 52660)
LD DE, 7D 00 (32000)
LD BC, CO 01 (448)
LDIR
RET
```

Basic Modifications

Modifications are required to Tasword Basic. There is not much spare room in the Basic area with Tasword loaded so some preliminary work has to be done. All the numbers in lines up to 1000 must be changed to VAL "number".

Provision must be made to display the information. One item on the STOP MENU has been changed and one has been added. Instead of "back to basic" there is "heading (for a letter)" and then "word count" has been added at the bottom of the menu.

1. Load Tasword in the normal way.

2. Edit every line to 1000 replacing numbers with VAL "number". Note that this doesn't apply to numbers in strings e.g. PRINT "2 - fix heading as typed", or numbers in variables e.g. j1, or initial line numbers but it does apply to GOTO and GOSUB line numbers. Each time you do this you save three bytes. You can check how much memory you have saved by typing in line 9990 and using GOTO 9990 every now and then. The new Basic requires over 1000 bytes of extra space. If you need Microdrive routines you will need to make even more space by using VAL "number" right through the program.

3. Add or modify the lines as shown in the listing, making absolutely certain that the

numbers in lines 9810 and 9820 are EXACTLY as printed, as a single error could crash the entire program.

4. Type GOTO 9800 and ENTER.

5. Now delete lines 9800 to 9990.

6. Save your new program on tape (or Microdrive) by using SAVE "tasword" LINE 15: SAVE "tasword" CODE 52610,12925 (or similar M/Drive commands).

7. Check that the program saved properly by VERIFYing it. Use VERIFY "" : VERIFY "" CODE.

The machine code is automatically called each time you go to the menu via SYMBOL SHIFT/STOP and again using option 't'. If you have made any mistakes at all then the program will crash and you will have to start again. Due to this it might be better to save the program after step 3 until you are sure it is working properly.

Once you have saved a copy as in steps 6 and 7 you can test it out by RUN. Going to the STOP menu you should see a word count of zero. Load a file or type

something in and take note of the wordcount value. If you have a fairly long text file, try out the paragraph count facility. You will need to note the start line and the end line of the paragraph you want to count then go to the STOP menu and select 'w'. Remember the routine will count separate punctuation marks like "." as complete words. The same will apply to numbers.

The Heading menu item allows you to print the heading already held at 52660 - or if there is no heading in then you can type your own in and 'fix' it to be recalled at any time. However, once you have fixed it you must then re-SAVE the program so that it will be available, each time you re-LOAD. You can do this saving by using item 't' on the menu.



```
25 GO SUB VAL "4000": PRINT AT
VAL "2",VAL "0";"print text fil
e
55 PRINT : PRINT "heading (for
letter)
60 PRINT : PRINT "word count t
otal = ";USR VAL "52610";TAB VAL
"31";"w"
170 IF b=VAL "104" THEN LET i=
VAL "18"
175 IF b=VAL "119" THEN LET i=
VAL "20"
180 IF i<>VAL "0" THEN PRINT A
T i-VAL "2",VAL "31"; FLASH VAL
"1";CHR$ b; GO TO VAL "500"
500 PRINT AT VAL "20",VAL "10";
: PRINT AT VAL "18
",VAL "30"; : PRINT #VAL "1";
press the "; FLASH VAL "1";"ENT
ER"; FLASH VAL "0";" key to proc
eed"" press "; FLASH VAL "1";"
c"; FLASH VAL "0";" to change ch
oice
670 IF b=VAL "104" THEN GO TO
VAL "9000"
680 IF b=VAL "119" THEN GO TO
VAL "9300"
699 REM delete
```

```
710 SAVE a$CODE VAL "52610",VAL
"12925": GO SUB VAL "900"
790 VERIFY a$CODE : PRINT AT VA
L "21",VAL "20";" m/code O.K. ":
RUN
9000 CLS : PRINT "new heading? y
/n"
9002 IF INKEY$(">")="y" AND INKEY$("<")
="n" THEN GO TO VAL "9002"
9010 IF INKEY$="n" THEN RANDOMI
ZE USR VAL "53134": RANDOMIZE US
R VAL "53110": RUN
9020 PRINT "1 - go back and type
new heading SEVEN LINES MAXI
MUM""2 - fix heading as typed"
9022 IF INKEY$(">")="1" AND INKEY$("<")
="2" THEN GO TO VAL "9022"
9030 IF INKEY$="1" THEN RUN
9040 RANDOMIZE USR VAL "53122":
RUN
9300 INPUT "Start line: ";x: IF
x>VAL "320" OR x<VAL "1" THEN G
O TO VAL "9300"
9310 PRINT AT VAL "18",VAL "31";
: " "; "para starts at line ";x;
9320 INPUT "End line: ";y: IF y>
VAL "320" OR y<VAL "1" OR y<x TH
```

```

EN GO TO VAL "9320"
9330 PRINT "para ends at line
";y
9340 LET pk=VAL "52614": LET st=
VAL "31999"+VAL "64"*(x-VAL "1")
: LET ln=VAL "64"*(y-x+VAL "1")
9350 POKE pk+VAL "3",VAL "256"*(
st/VAL "256"-INT (st/VAL "256"))
9351 POKE pk+VAL "4",INT (st/VAL
"256")
9352 POKE pk,VAL "256"*(ln/VAL "
256"-INT (ln/VAL "256"))
9353 POKE pk+VAL "1",INT (ln/VAL
"256")+VAL "1"
9360 PRINT "para count = ";USR V
AL "52610";TAB VAL "25"; FLASH V
AL "1";"any key": PAUSE VAL "0"
9370 POKE pk,VAL "0": POKE pk+VA
L "1",VAL "81": POKE pk+VAL "3",
VAL "255": POKE pk+VAL "4",VAL "
124": RUN
9800 RESTORE : READ a$: FOR n=52
610 TO 52657: LET a=VAL a$( TO 3
): POKE n,a: LET a$a=a$(4 TO ): N
EXT n
9810 DATA "00100000000170000081033
25512403502706200018620012625403

```

```

20402450030350270620001862000620
63165254000032007035126254032043
032235126254032032231024217"
9820 DATA "033180205017000125001
19200123717620103300012501718020
50011920012371762010330632030172
55204001064078237184201"
9840 READ a$: FOR n=53110 TO 531
45: LET a=VAL a$( TO 3): POKE n,
a: LET a$a=a$(4 TO ): NEXT n: STO
P
9990 PRINT VAL "65536"-USR VAL "
7962"

```



A FINAL WORD

MY NAME IS FRED STERN AND I AM
THE EDITOR OF THIS EDITION OF
LISTING.

SPECIAL THANKS GO TO BOB GILDER,
TOM SKAPINSKI AND DONALD LAMBERT
FOR THEIR CONTRIBUTIONS.

A VERY SPECIAL THANK YOU TO
HARVEY FOR HIS HOSPITALITY, AND
THE USE OF HIS STORE FOR OUR
MEETING. ALSO TO MIKEY FOR HIS
CONTRIBUTIONS.

SEE YOU ALL AT THE NEXT MEETING.



STAY

SOBER



FAST

FOOD



SHOE



TASWORD TWO has, justifiably, proved to be the most popular word processor program for the Spectrum. One of its attractive features is the ease with which you can modify it. The ideas which follow will allow you to customise Tasword in a number of ways and all will work on any version. They will also work if you have altered it to use Tasprint, the East London Robotics memory upgrade, microdrives and disc drives.

Throughout the article 'direct mode' denotes that commands can be entered directly from the keyboard and will give the OK report; 'Basic' is the Basic part of Tasword, reached by pressing STOP; and 'Tasword' denotes the code part of Tasword, reached by pressing Y from the STOP menu or RUN in direct mode. Sixty four mode denotes 64 characters per line and similarly 32 mode denotes 32 per line.

Changing the colours

As you might have discovered already, you can change the INK and PAPER colours used by Tasword simply by altering them in direct mode and running. That has disadvantages: the paper colour can only be 1, 3, 5 or 7 and the margins will be one less than that. To alter the colours in 64 mode enter, in direct mode

POKE 58512,54

POKE 58513,c

POKE 58521,54

POKE 58522,c

where c is $8 * \text{paper} + \text{ink}$. For example, to get blue paper with white ink use $c = (8 * 1 + 7) = 15$. To change the margin colour enter

POKE 58508,54

POKE 58509,c

where c is calculated as above. To change the border colour enter

POKE 64516,b

where b is between 0 and 7. To change the 32 mode border use

POKE 60641,b

calculating b as above.

Customising your word processor



John Lambert shows how to squeeze more from Tasword Two

Those POKEs will not change the bottom two lines. To change those

POKE 64570,c

and for the second to the bottom line use

POKE 59993,c

calculating c as above. Note that on this line the top half is INK and the bottom PAPER. (To keep the same colours as above you should poke it $7 * 8 + 1 = 57$.)

When returning to Basic the INK and PAPER colours are restored to those used before running Tasword but the Border is not. To remedy that add a BORDER command on line 20 before the CLS. As those POKEs are within Tasword they will be retained when the program is saved.

Lines, Characters, Words

When returning from Tasword, to the STOP menu, the program always goes

to line 20. When going to Tasword from the Basic part of the program there are two routes, from line 10 — the usual one — and line 3060 — in the process of using search and replace. That can be employed to add a count either of lines or characters used. To count lines add

79 PRINT #0;AT 1,0;a/64;" Lines Used - "; (PEEK 64150+256*PEEK 64151+22)-a/64;" Lines Free"

and to count characters add

79 PRINT #0;AT 1,0;ch;" Characters Used - "; ((PEEK 64150 + 256 * PEEK 64151 + 22) * 64) - ch; "Free"

Alter the USR statements in lines 10 and 3060 to be LET ch=USR, followed by the rest of the line, and add LET a=ch in line 20 after the CLS.

The fastest way to count words is to add a short machine code program. Enter listing one, run it and save the code on tape. Load a copy of Tasword and add the program lines in listing two — you will also have to alter the CLEAR address in line 15 to 31940 and add another LOAD CODE[™]. To ensure the code is saved automatically add

705 SAVE "word" CODE 31941,59
Alternatively provided you are only using Tasword from cassette add

705 POKE 23736,181: SAVE "word" CODE 31941,59: POKE 23736,181

Listing 1. word count data

```
10 LET clr=31940: LET t=0: FOR
n=1 TO 59: READ a: POKE (clr+n)
a: LET t=t+a: NEXT n: IF t<>640
5 THEN PRINT "ERROR": STOP
20 SAVE "word"CODE clr+1,59
97 REM *****
98 REM WORD DATA
99 REM *****
100 DATA 42,181,253,221,42,11,9
2,221,94,4,221,86,5,25,235,33,0,
0,1,0,64,27,26,254,32,40,24
110 DATA 254,128,48,20
120 DATA 14,1,16,242,9,58,182,2
53,186,32,232,58,181,253,187,32,
```

226,229,193,201,175,185,40,234,3
5,13,24,230

Listing 2. word count additional basic

```
1 DEF FN w(x)=USR 31941
15 SEE TEXT
174 IF b=119 THEN GO TO 4500
705 SEE TEXT
4500 CLS : GO SUB 4000: PRINT AT
11,5;"Number of Words = " : PRI
NT FN w(a)
4510 PRINT #0: " Press Any Key
To Continue": PAUSE 1: PAUSE 0:
GO TO 20
```

That will stop the 'press any key' message appearing and save the three parts but will cause odd results if used with microdrives or disc drives.

Once that is done LOAD the word code and then save TASWORD in the usual way.

The routine searches through the file, taking the length from Basic and the start of the file from locations within Tasword, and counts a word as being a non-space followed by a space. It includes words which start at the end of a line, making it a true count.

A space is taken to be a character with a code of 32 or a graphics character with a code of 128 or more. If you do not want to include those graphics then change the four DATA statements in line 110 to zero. If you use Tasprint the tilde character — code 126 — is used to change text styles and is treated by Tasword as a space. In that case change the second DATA item in line 110 (128) to 126.

The method of calling the routine may be new to you and relies on the way in which the Spectrum treats user-defined functions and stores numbers. When a function is called the system variable DEFADD holds the address of the definition of that function.

If a single letter is used the memory locations 3 to 7 on from that hold the five byte floating point form of the variable being used. If that value is an integer — between -32767 and 32767 — it can be found in locations 4 and 5. Unfortunately the Spectrum takes a relatively long time to delve into the ROM but the method is useful for routines which require one, or more, parameters to be passed.

Machine code buffs might like to move the word count code to a REM statement for ease of loading and saving.

If you are familiar with the calculator routines in the ROM, you can make use of this method in your own programs; the INT-STORE routine in the ROM at 2D8Ch is particularly useful. For further discussion of the calculator see *Helpline* in *Sinclair User*, February.

Graphics and Pounds

If you are using a full-sized printer you can send control codes to it using the graphics characters. It is a good idea to add a space — 32 — to the sequence as otherwise the line, when printed, appears to have lost a character. That can be added before or after

Listing 3. Assembly listing of Listing 1.

```
HL=Word Count
DE=Position in File
S=64 (Characters per Line)
C=Flag 0=Space 1=Non-Space

EA60 100 ORG 60000 ;Arbitrary ORG
FDB5 110 FILEST EQU 64949 ;Address in Tasword
                                that stores the
                                start of the file
                                ;System Variable
SC09 120 DEFADD EQU 23563 ;Get start of file
EA60 2AB5FD 130 INIT LD HL,(FILEST) ;Pass the value of
EA63 D02A0B5C 140 LD IX,(DEFADD) ;a (file-length) to
EA67 D05E04 150 LD E,(IX+4) ;DE
EA6A D05A05 160 LD D,(IX+5) ;HL=start + length
EA6D 19 170 ADD HL,DE ;Move it to DE
EA6E EB 180 EX DE,HL ;HL=Word Count
EA6F 210000 190 LD HL,0 ;Load S,64 & C,0
EA72 010040 200 NXLINE LD BC,64*256+0 ;Step through file
EA75 1B 210 NXCHAR DEC DE
EA76 1A 220 LD A,(DE)
EA77 FE20 230 CP 32 ;if Space -
EA79 2B18 240 JR Z,FNDSPC ;jump
EA7B FE90 250 CP 128 ;if Graphic -
EA7D 3014 260 JR NC,FNDSPC ;jump
EA7F 0E01 270 LD C,1 ;set flag
EAB1 10F2 280 LINEND DJNZ NXCHAR ;if not end of line
                                get next char
                                ;flag holds state of
                                first char in line,
                                add to count
EAB3 09 290 ADD HL,BC ;find if address
                                ;counter (DE) has
                                ;reached the
                                ;start of the
                                ;file - if so
                                ;then get next line
                                ;move word count
                                ;to BC
                                ;and return
                                ;clear A
                                ;is flag 0
                                ;if so jump
                                ;increase word count
                                ;zero flag
                                ;back to routine
EA84 3AB5FD 300 LD A,(FILEST+1)
EA87 BA 310 CP D
EA89 20E8 320 JR NZ,NXLINE
EAB8 3AB5FD 330 LD A,(FILEST)
EA8D BB 340 CP E
EABE 20E2 350 JR NZ,NXLINE
EA90 E5 360 PUSH HL
EA91 C1 370 POP BC
EA92 C9 380 RET
EA93 AF 390 FNDSPC XOR A
EA94 B9 400 CP C
EA95 2BEA 410 JR Z,LINEND
EA97 23 420 INC HL
EA98 00 430 DEC C
EA99 1BE6 440 JR LINEND
```

the sequence. As a matter of personal preference I tend to put it after, except for the underline codes. Code 140 should have it before and 131 after.

The pound sign has caused many headaches. The problem is that, as far as the Spectrum is concerned, the pound has a code of 96 whereas most printers use a code of 35. To further complicate matters the printer also uses 35 to denote the hash character.

It does that by having a number of alternative character sets and you have to switch between them, either using software or by setting switches in the printer. The code of 96 will be printed as a backward apostrophe.

The ideal solution would be to check all the characters being sent to the printer and make the necessary changes. This can be done in software but as it will vary according to the printer and interface used it is difficult to give a general method.

Two methods can be used. The first is to set the printer to use the USA character set so that it will print a hash when sent a code of 35. In that way the hash in listings will be printed correctly.

If one of the graphics characters — say 143 — is set to give the English character set and 128 to give the USA set you can alternate between them at will. You must however remember

whether the hash on screen will print as a pound or as a hash.

If you rarely use the hash character the other method will suffice. Set the printer to the English character set and alter the hash character so that it looks like a pound sign on screen. It will still have to be obtained by pressing Symbol Shift 3 but at least you can see what is happening. To do that enter, as a direct command

```
FOR n = 0 TO 7: POKE (61208 + n), PEEK (61696 + n): NEXT n
```

The pound sign on the keyboard will still give you a pound on screen so enter

```
FOR n = 0 to 7: POKE (61696 + n), 0: NEXT n: POKE 61697,4: POKE 61698,2
```

to change it to a backward apostrophe. Now you have only to remember to press the correct keys but at least what you see is what you get. If you want to use hash you will have to use the graphics characters to swap sets as before.

Help Pages

If you change the uses of the graphics characters you will need to alter the help pages accordingly. The easiest way is to move them into the text file to work on. Enter in direct mode

```
LET s = FN p(64949): FOR n = 0
```


TO 1407: POKE (s + n), PEEK (54784 + n): NEXT n which will move it into the file. To return it to its correct place — provided you have not moved it from the start of the file — go into direct mode and enter

LET s = FN p(64949): FOR n = 0 TO 1407: POKE (54784 + n), PEEK (s + n): NEXT n

To work on the Extended mode page replace the 54784 in the above with 56320.

Keyboard

Tasword uses two of the system variables to time its reading of the keyboard. REPDEL at 23561 stores the length of time a key must be held down before it repeats and is normally set to 35. That can be usefully changed to around 20 for fast typists.

One useful modification is to alter the sound made when a key is pressed, as on a rattly keyboard the original sound is usually lost. The BEEPER routine in the ROM at 03B5h is used which needs, on entry, the DE register holding the frequency \times time and HL holding the length of the timing loop.

Tasword sets D to zero and E to the value of PIP (23609), normally 5 but set to 2 by line 15. The value of HL is held in 59161 and 59162; POKEing 59161,50 gives a loud(ish) typewriter click.

Headers, footers...

Listing four is a routine which allows you to add headers, footers and page numbers to a printed text file, and print as many copies as you want. Delete lines 200 to 300 from your original copy of Tasword and type in the listing. Due to memory limitations, headers and footers should be included in the listing as shown.

Lines 200 to 240 should be self-explanatory as they ask for input from the user. Line 250 calculates which part of the text file is to be printed and tells you, on screen, what is happening. Lines 255 and 260 are more calculation. Line 270 puts your header here — remember to allow for the margin. Line 275 is part of the original Tasword Basic and takes into account different printer interfaces; you may find that causes a blank line to be output.

Line 280 calls Tasword to do the printing. Line 285 is similar to 275. Line 290 checks if you want to quit printing; the RUN ensures that all the

additional variables created are CLEARed. Line 292 ensures that the page number is printed at the bottom of the page. Line 294 puts your footer here. Line 296 prints the page number; as most interfaces do not handle TAB correctly a number of spaces are LPRINTed. The number is positioned in the middle of the page. PEEK 60927 is the width of the margin.

The routine needs one memory location to store the number of text lines per page and I have used 31940 for that. You will have to alter the CLEAR in line 15 to 31939. Alternatively, you could use one of the unused locations in the system variables, 23681, 23728 or 23729, provided your printer interface does not use it.

The routine has been made as short as possible but if you use a microdrive or the Thurnall, Stacom or Watford disc systems you may find you get an 'out of memory' error. That is because those systems use memory when loading or saving. Using cassette, or the Technology Research disc system, does not.

Try the routine using a file which you have safely saved. If you do get the 'out of memory' report you will have to shorten the Tasword Basic. Two methods can be used, both of which are used in the routine. The first is to change all the numbers to use VAL — that is 123 becomes VAL "123" and so on — and that will save three bytes per number. It may not seem a great deal but in the routine 200 bytes are saved by this method.

The second method is to use variables in place of numbers. In the routine z is used for zero and u for one. If you add LET z=0: Let u=1 to line 10 and then change all occurrences — omitting line 15 and the subroutine at 4000 — you can save a further three bytes every time.

Depending on your version of Tasword you can save up to 1.5K which is more than enough for this routine and any of your own. As long as you have 700–800 bytes free you should not have any problems — to check the amount of free memory enter

PRINT 65535—USR 7962

Finally...

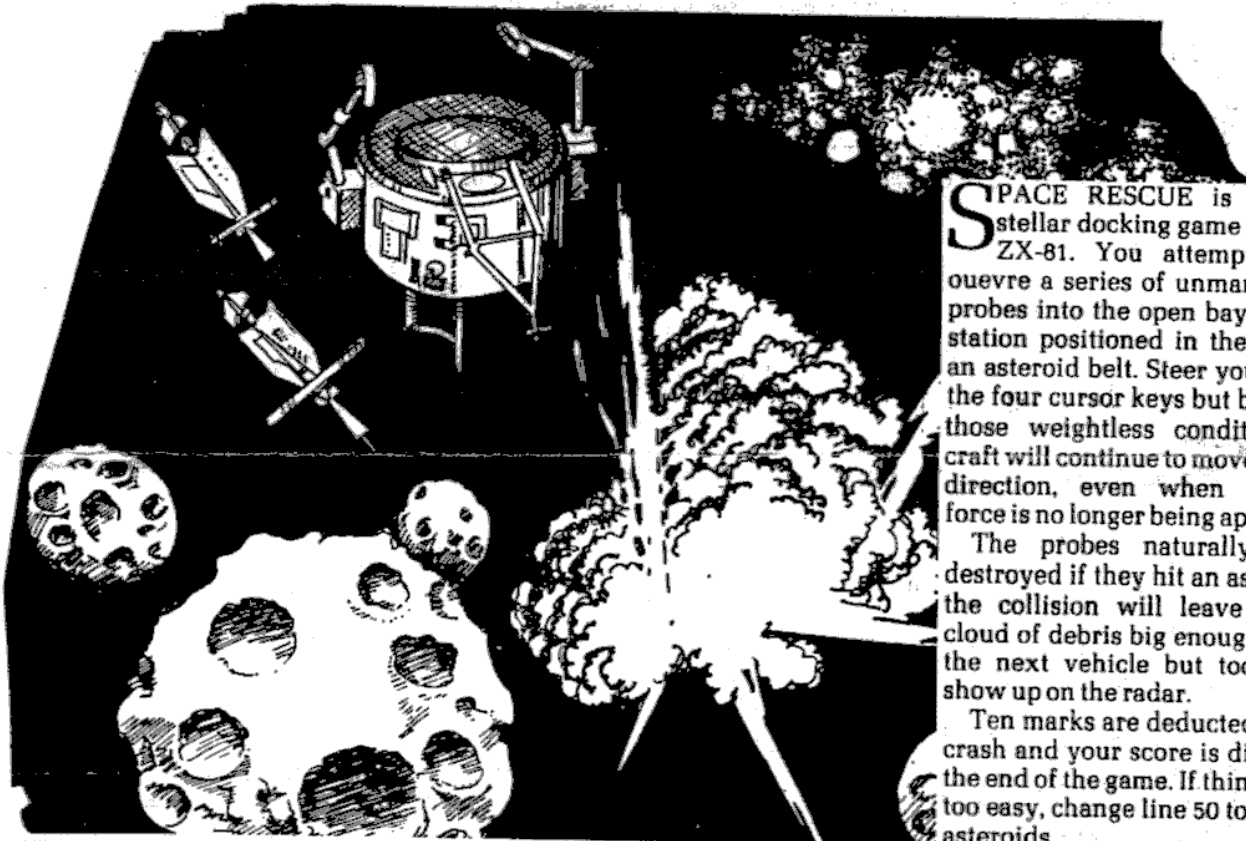
All the above will prove useful and enhance what is already an impressive program. If you have further ideas for improvements I would be pleased to hear from you c/o Sinclair User.

Listing 4. Print routine for headers, footers, page numbers and multiple copies

```
200>CLS: GO SUB VAL "4000": PR
INT AT VAL "4",VAL "8":PRINT OP
TIONS: PRINT " just press ENTE
R for default values given in
brackets": LET z=VAL "0": LET u
=VAL "1": LET gs=VAL "6000": LET
i=VAL "8": LET j=z: LET jo=VAL
"23": LET x=VAL "31940": LET a$=
"Text Lines per Page": GO SUB V
AL "850": LET i=VAL "10": PRINT
AT i,z:"Line Spacing? (1)": GO S
UB gs: IF a$="" THEN LET a$="1"
210 POKE VAL "62235",VAL a$: LE
T i=VAL "12": PRINT AT i,z:"Star
t at line? (1)": GO SUB gs: IF a
$="" THEN LET a$="1"
220 LET c=VAL "64"*(INT VAL a$-
u): LET st=c+FN p(VAL "62216"):
LET i=VAL "14": PRINT AT i,z:"Fi
nish at Line? (last)": GO SUB gs
: IF a$="" THEN LET len=a-c: GO
TO VAL "230"
225 LET len=VAL "64"*(INT VAL a$
-c
230 LET i=VAL "16": PRINT AT i,
z:"First Page Number (1)": GO SU
B gs: IF a$="" THEN LET a$="1"
240 LET pa=VAL a$-u: LET i=VAL
"18": PRINT AT i,z:"Number of Co
pies (1)": GO SUB gs: IF a$="" T
HEN LET a$="1"
250 LET co=VAL a$: LET end=st+1
en: LET pl=INT (PEEK VAL "31940"
/PEEK VAL "62235")*VAL "64": LET
tp=INT ((len/pl)+u): FOR neu TO
co: LET fl=z: LET cp=pa: FOR f=
st TO end-u STEP pl: LET b=f: LE
T x=VAL "60045": GO SUB VAL "950
": LET cp=cp+u: RANDOMIZE USR VA
L "59806": RANDOMIZE USR (FN p(V
AL "62472")): CLS: PRINT AT VAL
"10",z:"Press the q key to quit
printing": AT VAL "14",z:"Printi
ng Copy Number": AT VAL "18",z:
"Page Number ":cp: IF end-f<pl T
HEN LET b=end-f: LET fl=u: GO TO
260
255 LET b=pl
260 LET x=VAL "60049": GO SUB V
AL "950"
270 REM LPRINT "Header"
275 LET c=PEEK VAL "62470": IF
c<>z THEN LPRINT CHR$ c
280 RANDOMIZE USR VAL "60038"
285 LET c=PEEK VAL "62471": IF
c<>z THEN LPRINT CHR$ c
290 RANDOMIZE USR VAL "59806":
IF INKEY$="q" THEN RUN
292 IF fl=u THEN FOR g=u TO (pl
-b)*PEEK VAL "62235"/VAL "64": L
PRINT: NEXT g
294 REM LPRINT "Footer"
296 FOR g=1 TO 25+PEEK 60927: L
PRINT " ": NEXT g: LPRINT "Page
No ":cp: of ":tp: LPRINT CHR$
12: NEXT f: NEXT n: RUN
Variables used in Listing 4.
st = start of file
len = length of the file
pa = first page number
co = number of copies
end = end of file
pl = length of page (to be printed)
tp = total number of pages
fl = flag
cp = current page
```

There is one problem, at least, to which I do not have the answer. The length of the text file (–22) is held in locations 64150 and 64151. Try POKEing 64150,122, RUN Tasword, clear the text file — E Mode X — and scroll through it. How do you get rid of line 357?

SPACE RESCUE - T31000



SPACE RESCUE is a kind of stellar docking game for the 16K ZX-81. You attempt to manoeuvre a series of unmanned space probes into the open bay of a space station positioned in the middle of an asteroid belt. Steer yourself with the four cursor keys but beware—in those weightless conditions your craft will continue to move in a given direction, even when propulsive force is no longer being applied.

The probes naturally will be destroyed if they hit an asteroid and the collision will leave behind a cloud of debris big enough to wreck the next vehicle but too small to show up on the radar.

Ten marks are deducted for every crash and your score is displayed at the end of the game. If things become too easy, change line 50 to give more asteroids.

```

1 LET U=0
10 DIM A(40,40)
15 FAST
20 FOR F=0 TO 21
30 PRINT "
40 NEXT F
50 FOR N=0 TO 30
60 LET X=INT (RND*30)+1
70 LET Y=INT (RND*20)+1
80 LET A(X,Y)=1
90 PRINT AT X,Y:CHR$ 130
100 NEXT N
110 GOSUB 5000
111 SLOW
120 LET Y=0
130 LET R=30
140 LET E=15
150 LET A=CODE INKEY$
160 IF INKEY$="" THEN LET A=Y
170 LET Y=R
180 PRINT AT R,E:" "
190 IF A=33 AND E>0 AND E<30 TH
EN LET E=E-1
200 IF A=35 AND E>0 AND E<30 TH
EN LET E=E+1
210 IF A=35 AND R>0 AND R<21 TH
EN LET R=R-1
220 IF A=34 AND R<21 AND R>0 TH
EN LET R=R+1
230 LET U=U+1
240 LET S=A(E,R)
250 PRINT AT R,E:" "
260 IF S=1 THEN GOSUB 1000
270 IF S=10 THEN GOSUB 2000
280 GOTO 200
1000 PRINT AT 0,0:"SCORE=";U-10
1010 LET U=U-10
1020 PRINT AT R,E:" "
1030 PAUSE 50
1040 PRINT AT R,E:" "
1050 GOTO 170

```

```

2000 FOR F=0 TO 20
2010 PRINT AT 0,0:"CONGRATULATIO
NS"
2020 PRINT AT 0,0:"CONGRATULATIO
NS"
2030 NEXT F
2040 PRINT AT 0,0:"SCORE=";U+100
2050 PAUSE 100
2055 CLS
2060 GOTO 10
4999 STOP
5000 LET U=INT (RND*20)+7
5010 LET O=INT (RND*10)+7
5020 LET A(U,O)=10
5030 PRINT AT O,U:" "
5040 FOR F=U-2 TO U+2
5050 PRINT AT O+2,F:" "
5055 LET A(O+2,F)=1
5060 PRINT AT O-2,F:" "
5065 LET A(O-2,F)=1
5070 NEXT F
5080 FOR F=O-2 TO O+2
5090 PRINT AT F,U-2:" "
5095 LET A(F,U-2)=1
5100 PRINT AT F,U+2:" "
5105 LET A(F,U+2)=1
5110 NEXT F
5120 LET T=(INT (RND*4))
5130 IF T=0 THEN PRINT AT 0,U+2:
" "
5140 IF T=0 THEN LET A(0,U+2)=0
5150 IF T=1 THEN PRINT AT 0,U-2:
" "
5160 IF T=1 THEN LET A(0,U-2)=0
5170 IF T=2 THEN PRINT AT 0+2,U:
" "
5180 IF T=2 THEN LET A(0+2,U)=0
5190 IF T=3 THEN PRINT AT 0-2,U:
" "
5200 IF T=3 THEN LET A(0-2,U)=0
5210 RETURN

```